June 4, 2021

GNU/Linux OS I/Q recording program for AOR AR5700D as well as AR2300 / AR5001D / AR6000 equipped with the IQ5001 option.

Instructions for data version “rel_210601”. Some instructions have changed, please read this manual again thoroughly.

Sound and spectrum quality improved compared to previous release “rel_210203”.

Confirmed to work on:

Raspberry Pi O.S on RPI400/RPI4B

UBUNTU 20.10 on X86(Intel) 64bit PC

It is likely to work on most modern Linux based hardware, with recent Linux distributions, as long as the packages described section 1.2 are installed. However, some systems might require adjustments. The best way to know is to try it out yourself!

Confirmed not to work on:

Raspberry Pi O.S on Raspberry Pi zero,1/2/3 series

Required Linux knowledge: Medium to advanced

You must be comfortable with Linux operating systems, know how to install software packages, manage folder permissions and compile data.

AR5700D, AR2300 / AR5001D / AR6000 + IQ5001 will simply be described as "receiver".

<supplied files>

* IQ_for_Linux_English_instructions.txt --- This file

* 98-iqfk.rules --- File for udev

* Makefile --- Used to compile ariq_rcv.cc

* ariq_rcv.cc --- The source code of the recording program

* iqwr.sh --- Firmware download executable file

* fx2fw.hex --- Firmware file

* iq5001_file_in.grc, iq5001_file_in.py, grc-sample.png --- Sample GNU Radio files (Minimum required version to load this grc file is 3.8)

* iq5001_file_in_RPI.grc, iq5001_file_in_RPI.py, grc-sample_RPI.png --- Sample GNU Radio files for version 3.7 (for example on a Raspberry Pi which still uses v.3.7)

<Features>

* By using this program, I/Q data can be saved on a system running GNU / Linux OS. The data is stored as a “.bin” file. Format: 32bit floating point (numpy.float32)

* The recorded files can be used with GNU Radio.

* The recorded files can be played back with GQRX (some limitations apply).

* The source code is released under GPLv3 (or a later version), so anyone can use it freely.

<How to use>

1. Preparations

1.1

Have an updated GNU / Linux system ready to use.

1.2

Install following packages:

cycfx2prog - Cypress EZ-USB FX2 (LP) programmer

usbutils - Linux USB utilities

libusb-1.0-0 - userspace USB programming library

libusb-1.0-0-dev (required to compile files)

clang - Compiler

gnuradio (v3.7 or newer)

1.3

Copy 98-iqfk.rules to

/etc/udev/rules.d/

Execute the following command in terminal:

```
$ sudo udevadm control -R
```

1.4

Power on the receiver, connect the I/Q output USB cable to the PC. (The receiver's second USB connection for receiver control is not required)

Run the following command to download the firmware to the receiver.

```
$ source iqwr.sh
```

REMEMBER: This command must be executed every time the receiver is turned on!

1.5

ariq_rcv.cc needs to be compiled on each PC hardware/operating system you want to use.

Execute the following command in terminal:

```
$ make clean
$ make
```

The resulting ariq_rcv is now ready to use.

2. Record I/Q data

Copy the file ariq_rcv to the directory where you want to save your I/Q data.

Important note for RPI400 and RPI4B users: Due to SD card WRITE speed limitations and the receiver I/Q being output at 72Mbit/sec (9Mbyte/sec), be sure to save the I/Q stream to an external USB drive connected via USB 3.0. SD cards are just too slow and are very likely to create capture errors and/or abnormal audio and spectrum playback.

The following command will start the recording.

```
$ ./ariq_rcv
```

This will create the I/Q capture file "iqdatIQ.bin".

If it starts normally, the following 5 lines will be displayed while it is capturing data:

```
Version: rel210601-0f

Send capture start command.

6

FirstAbandoned

FirstContacted, bf=2
```

Note on the bf value: It can be 0,2,4,6 and informs data leakage. 0 is the least leaked data, and 2,4,6 means some higher leakage.

The cause of the leak is unknown at this time but it should have little impact for GQRX or GNU Radio use. This bf number changes when the receiver is powered on, and it has also been confirmed that it changes from time to time by repeating the end and start of data reception.

To stop the capture, press Ctrl-C. (Alternatively you can also send SIGINT, SIGTERM or SIGQUIT signals to ariq_rcv.

If successful, the program will exit after the following eight lines are displayed:

```
Capture done.

6

Sent capture end command.

canceled

freed

released

closed

exit
```

3. Q&A

Q.

The capture file cannot be generated. Why?

A.

There are several possible causes.

1. The firmware may not have been downloaded to the receiver. The receiver being switched on and the I/Q output cable being connected to the PC, enter the following command:

```
$ lsusb -d 08d0:a001
```

The command's output should be as follows:

```
Bus 001 Device 007: ID 08d0:a001 AOR, LTD. DIGI-RECEIVER
```

(Bus and Device number will depend on your system and are not important here.)

However, if "AOR, LTD. DIGI-RECEIVER" is missing, then the firmware download has failed.

2. If you entered the following command:

```
$ lsusb -d 08d0: a001
```

and there is no response at all, then the system is unaware of the receiver. Check the USB cable connection and confirm that the receiver is turned on.

3. The udev settings (step 1.3) may not have completed successfully. Please check.

4. Check if the permissions of the directory where you are recording the file are correct.

Q.

How much data is the AOR I/Q file?

A.

The sampling rate is 1.125MSps and the data length per sample is 64 bits. The transfer rate is 9 Mbytes / sec (72 Mbit / sec). In other words, the I and Q files will be 270MB per minute each, and the combined I/Q file 540MB per minute.

Q.

Can it be run on a Raspberry Pi?

A.

Yes. However, only on the RPI400 and RPI 4B. Also, as SD card WRITE speeds are too slow, I/Q data must be saved to an external USB memory or drive, connected via USB 3.0.

Q.

Can it be run on Windows?

A.

No. However our receivers featuring I/Q output are supplied with the AR-IQ3 Windows software which allows I/Q capture and playback.

Q.

Can I use libusb-0.1 instead of libusb-1.0-0?

A.

No because libusb-0.1 does not support isochronous transfers.

Q.

Can I rewrite the program in Python?

A.

There is a libusb-1.0-0 wrapper for Python3.

Q.

Why can't I use a "live" I/Q stream in GNU Radio, only a captured file?

A.

We are considering it in a future version. Check http://www.aorja.com/support/software.html from time to time.

Q.

What is the "device string" to use in GQRX?

A.

Device= Other

file=(path to your iqdatIQ.bin file),rate=1.125e6,repeat=true,throttle=true

Q.

I recorded I/Q on H.F, but when played back in GQRX or GNU Radio, there seem to be no signal. Why is that?

A.

Remember that as described in the AR-IQ3 manual, when using the I/Q stream and receiving frequencies below 25MHz, the HF antenna should be physically connected to the antenna socket #1, even though it’s usually #2 when the receiver is used “stand-alone”. This is due to the fact that the receiver’s internal antenna wiring is different when used “stand-alone” vs controlled by software. It also means that for pure I/Q capture you won’t be able to hear via the receiver what you are receiving. A workaround is to connect an antenna to both socket #1 and socket#2, during the capture.

4. Known limitations in GQRX

-The displayed center frequency is zero, which is incorrect. However, you can change it manually via the digits.

-The receive frequency cannot be changed via the digits, only by dragging the vertical bar on the spectrum with your mouse.

AOR Ltd.

www.aorja.com